

REMARKS

In the Office Action dated May 26, 2005, the Examiner rejected claims 1-5, 7-15, and 17-20 under 35 USC 103(a) as unpatentable of Raj (Enterprise JavaBeans) in view of Jain (Java Call Control, Coordination and Transactions, IEEE Communications Magazine, January 2000) and rejected claims 6 and 16 under 35 USC 103 as unpatentable over Raj, Jain, and Shoffner (Write a Session EJB). Claims 1 through 20 remain at issue.

The Art Rejections

The Applicant would like to first address the comments made by the Examiner in the "Response to Arguments" (paragraph 20) in the current Office Action.

In the "first point", the Examiner states the applicant was incorrect when arguing that Raj fails to teach or suggest the remote interface associated with an entity bean class. The Applicants disagree with the Examiner's conclusion. A careful review of the actual statements made by the Applicants demonstrate that the Examiner has conveniently mischaracterized what was said. Specifically, the Applicants stated *"As noted by the Examiner, Raj fails to teach or suggest a remote interface associated with the entity bean class, the remote interface being arranged to drive the state machine in response to input events"* (emphasis added in bold). Raj in fact does not teach a remote interface that is arranged to drive a state machine in response to input events. The Examiner acknowledged this very point in the first Office Action dated February 9, 2005. See specifically paragraph 6 where the Examiner stated *"However, Raj does not teach the remote interface being arranged to driver the state machine"* The Examiner made the exact same statement in paragraph 5 of the current Office Action. The first premise used by the Examiner for finding the Applicants remarks non-persuasive is therefore based on an incorrect characterization of the Applicant's statements.

In the second point, the Examiner attempts to explain the applicability of the Jain reference. In so doing, the Examiner noted several references to a Finite State Machine (FSM) and Enterprise Java Beans (EJBs) that appear in the Jain reference. A careful review of the Jain demonstrates that the reference is irrelevant to the present invention and does not teach the features missing from the Raj reference.

In paragraph 5 of the current Office Action, the Examiner attempts to characterized Jain as teaching a finite state machine implemented as a Java object. As previously noted, however, the finite state machine of Jain is for handling telephone calls and connections. There is no teaching in Jain whatsoever of using a finite state machine as a Java object. The listing of examples where Jain mentions finite state machines or FSMs at various locations in the reference are therefore irrelevant to the present invention. The use of a state machine to handle telephone calls and connections has absolutely nothing to do with implementing a finite state machine as a Java object or supporting enterprise platforms as recited in the claims of the present invention.

The Examiner has also mischaracterizes the Jain reference in construing the rejection. In the same paragraph 5, the Examiner states the "...*The state of a call ... to the application; page 110, right column, second paragraph), could be implemented using Enterprise Java Bean (Trusted application ... EJB sense; page 112, left column, third paragraph).* A review of page 112, left column, third paragraph, indicates the use of Enterprise Java Beans (EJBs) has nothing to do with implementing a call to the state machine. Rather, the reference states:

Note in the diagram (figure 2) we show the JSLEE (Java Service Logic Execution Environment) residing between applications and JCC/JCAT (Java Call Control)/Java Coordination and Transactions). Obviously, trusted applications would actually execute using the facilities of the JSLEE, which would act as a "container" in the EJB sense. (explanation of the various acronyms provided in bold for convenience)

The above paragraph of Jain is simply defining the approximate relationship of the various JAIN edit groups. See the footnote under Figure 2 on page 111 of Jain. There is no teaching in the reference regarding the use of EJBs for implementing calls to a state machine.

Again, the Jain (and the JTAPI specification) fails to teach or suggest the very features that the Examiner is relying on in combination with Raj to reject the claims of the present application. Specifically, as noted in the previous response and repeated herein for convenience, the Jain reference teaches:

The finite state machines mentioned in the Jain reference are for calls, connections, etc., respectively. In contrast, the present invention relates to

implementing state machines as enterprise beans on an enterprise platform. The state machine is used within a computing environment that supports an enterprise platform that includes an entity bean class, a home interface associated with the bean class, and a remote interface that is associated with the entity bean class. The entity bean class, the home interface, and the remote interface collectively implement entity objects. Further, with the present invention, the state machine corresponds to an entity object. The home interface is used to create, find, and remove state machines, while the remote interface is used to drive the state machine transitions. The entity objects, entity bean class, the home interface, and the remote interface are realized as an Enterprise JavaBean entity bean component. Accordingly, the Jain reference (and the JTA API specification) fail to teach or suggest:

*a remote interface associated with the entity bean class,
the remote interface being arranged to drive the state machine in
response to the input events;*

The Jain reference thus fails to teach or suggest a remote interface arranged to drive a state machine in response to input events. Since neither the Raj or Jain references teach or suggest, among other elements of the present invention, a remote interface being arranged to drive the state machine in response to input events, the claims of the present invention are therefore allowable.

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

James W. Rose
Reg. No. 34,239

P.O. Box 70250
Oakland, CA 94612-0250
(650) 961-8300